European Congress on Computational Methods
in Applied Sciences and Engineering (ECCOMAS 2012)
J. Eberhardsteiner et.al. (eds.)
Vienna, Austria, September 10-14, 2012

ADJOINT MESH DEFORMATION AND ADJOINT-BASED SENSITVITITIES WITH RESPECT TO BOUNDARY VALUES

Anna Engels-Putzka[1] **and Christian Frey**[1]

[1] German Aerospace Center (DLR), Institute of Propulsion Technology
Linder Höhe, 51147 Cologne, Germany
e-mail: {anna.engels-putzka,christian.frey}@dlr.de

Keywords: adjoint methods, elliptic mesh deformation, mesh sensitivity, boundary conditions

Abstract. *This paper presents recent developments related to an adjoint RANS solver, which is part of the TRACE flow solver developed at DLR in particular for applications in turbomachinery. On the one hand, the implementation of an adjoint preprocess is described, which consists of two separate parts. First, the adjoint solution is transformed into a field of sensitivities with respect to grid point coordinates by calculating the derivative of the flow equation residual with respect to these coordinates. Second, an elliptic grid deformation procedure is adjoined in order to translate these sensitivities to surface values. The aim is to reduce the computational cost for sensitivity evaluations in applications with a large number of parameters by avoiding the generation of deformed volume grids. On the other hand, the evaluation of sensitivities with respect to boundary values, i.e. average quantities (for example pressure) which are prescribed at entry and exit surfaces, is discussed. In this context we also discuss an extension of the exit boundary conditions for the adjoint and linear solvers, which corresponds to the radial equilibrium condition in the nonlinear solver. For all these developments, exemplary applications to turbomachinery configurations are presented.*

1 INTRODUCTION

For many applications of Computational Fluid Dynamics (CFD) it is important to know not only global values computed from the flow solution (e.g. mass flow, pressure ratio, or efficiency) for a given configuration, but also sensitivities of such functions with respect to various parameters. In particular this is true for aerodynamic design and optimization [1, 2], where many different geometries have to be taken into account. Many optimization procedures make use of gradients of the objective functions. This information can be used either directly in the optimization algorithm or for the training of surrogate models [3, 4, 5].

A straightforward way to compute such gradients is by finite differences, i.e. the evaluation of the objective function for two configurations which differ by a small perturbation in the parameter of interest. But this approach has two major drawbacks. First, it is often difficult to determine the optimal step size for the perturbation to avoid either large truncation or cancellation errors. Moreover, one needs as many function evaluations – and therefore also solutions of the flow equations – as there are parameters. For complex optimization problems with hundred or more parameters this can become very costly.

Alternatively, the adjoint approach can be used, which is based on optimal control theory [6]. It was introduced in the context of CFD by Jameson [7]. The main advantage is that the adjoint problem has to be solved only once for each objective function, while the computation of sensitivities for arbitrary parameters from the obtained adjoint solution is relatively cheap. Since solving the adjoint problem has approximately the same complexity as one solution of the flow equations, the adjoint approach yields a speed-up with respect to the forward approach if the number of parameters is significantly larger than that of objective functions taken into account. Besides aerodynamic optimization, applications of the adjoint method include the examination of manufacturing and wear-and-tear tolerances [8] as well as error estimates [9, 10, 11].

The evaluation of an objective function depending on a parameter α usually consists of the successive application of several tools. For the case of variations of the blade geometry this process chain is illustrated in Fig. 1. Adjoining such a process chain as a whole is discussed for example in [12]. To evaluate the derivative of the functional with respect to α exactly one has to apply the chain rule. If, for example, $I(\alpha) = f(g(h(\alpha)))$, it is

$$\frac{dI}{d\alpha} = \frac{df}{dx}\frac{dg}{dy}\frac{dh}{d\alpha}. \tag{1}$$

If all steps are adjoined, the derivative is calculated only once for each functional and can then be used to obtain the variation of this functional from an arbitrary variation in α. But there are also other possibilities to evaluate the product in Eqn. (1). If, for example, the last function h is not included in the adjoint process, we have

$$\frac{dI}{d\alpha} = \left(\frac{df}{dx}\frac{dg}{dy}\right)\frac{dh}{d\alpha}, \tag{2}$$

where the part in parenthesis is evaluated only once (in adjoint mode), while $\frac{dh}{d\alpha}$ has to be calculated in forward mode for each value of α.

In this paper we present extensions of the CFD software package TRACE developed at DLR [13, 14], which contains besides solution methods for the nonlinear (steady or unsteady) RANS equations also solvers for the corresponding linearized and adjoint equations. The adjoint solver [15] uses a discrete adjoint approach. In the linear solver [16] the linearized equations are solved in frequency domain. For zero frequency this corresponds to solving the linearized steady RANS equations for small stationary disturbances. This can be used to validate

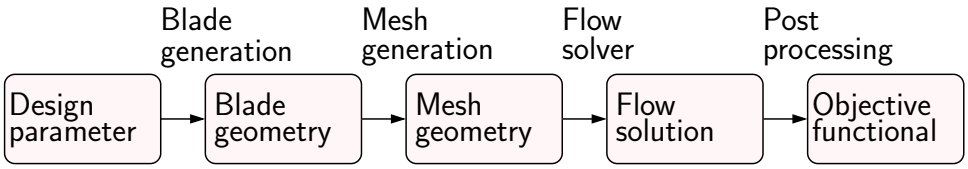

Figure 1: Schematic representation of a process chain for sensitivity evaluation.

results of the adjoint solver. Both solvers employ a preconditioned GMRES algorithm with restarts for the solution of the resulting linear equation system.

In the framework of TRACE so far only the flow solver itself and the post-processing (for certain functionals) have been adjoined [15, 17]. The first part of the process chain (pre-process) is evaluated in forward mode, which means that for each parameter variation a new computational grid is generated. Apart from the computational effort, in applications with large numbers of parameters also the storage space needed for these deformed grids can become an issue. Therefore it is desirable to adjoin also (parts of) the pre-process. Here we discuss the step from the surface geometry – more precisely, a surface grid – to the three-dimensional computational grid. Its adjoint is implemented as a post-processing of the adjoint solution. As a starting point we do not use the mesh generation process itself, but an elliptic mesh deformation [18, 19] which is applied to the computational grid for the initial geometry. This is discussed in detail in Section 2.

In the second part of the paper (Section 3) we deal with another extension of the adjoint solver, namely the evaluation of sensitivities with respect to parameters which are not variations of the geometry, but of boundary values. At artificial (open) boundaries of the computational domain certain flow parameters have to be prescribed, and the solution depends of course on the values of these parameters. In this work we consider in particular the nonreflecting boundary conditions implemented in TRACE for entry and exit surfaces. The knowledge of sensitivities of objective functionals with respect to these boundary values is useful for applications like aerodynamic optimization. While a fixed exit pressure is usually used as boundary condition in the flow solver, one is often interested in performing calculations for a fixed mass flow, which can be achieved by employing a mass flow controller. Adjoining such a controller can be avoided if sensitivities with respect to exit pressure can be efficiently calculated.

In turbomachinery applications it is also common to use the so called radial equilibrium condition at outlets. This means that the distribution of the static pressure over the radius is not prescribed explicitly, but only the pressure at one point (e.g. at midspan). The radial distribution is then given as (approximate) solution of an ordinary differential equation. To get consistent results with the nonlinear solver it is necessary to have an analogous boundary condition also in the linear and adjoint solvers. The implementation, which uses similar ideas as the inhomogeneous boundary conditions employed for the sensitivity calculation, is described in Section 3.3.

2 ADJOINT MESH DEFORMATION AND MESH SENSITIVITIES

The sensitivity of a functional I with respect to a (geometric) parameter α is given by

$$\frac{dI}{d\alpha} = \frac{\partial I}{\partial q}\frac{dq}{d\alpha}, \tag{3}$$

where we assume $I = I(q(\alpha))$, i.e. the functional depends on α only through the flow field q. If q is determined by the condition $R(q, x) = 0$, where R is the residual of the discretized flow equations and x represents the computational grid, we also have

$$0 = \frac{d}{d\alpha} R(q(\alpha), x(\alpha)) = \frac{\partial R}{\partial q}\frac{dq}{d\alpha} + \frac{\partial R}{\partial x}\frac{dx}{d\alpha} \tag{4}$$

and thus $\frac{dq}{d\alpha}$ can be computed as solution of the linear problem

$$\frac{\partial R}{\partial q}\frac{dq}{d\alpha} = -\frac{\partial R}{\partial x}\frac{dx}{d\alpha}. \tag{5}$$

If many different parameters α_j are of interest, this approach is not very efficient, since a system of the form (5) has to be solved for each parameter. Alternatively, the adjoint approach can be used, leading to

$$\frac{dI}{d\alpha} = -\psi^t \frac{\partial R}{\partial x}\frac{dx}{d\alpha}, \tag{6}$$

where the adjoint solution ψ is given by

$$\left(\frac{\partial R}{\partial q}\right)^t \psi = \left(\frac{\partial I}{\partial q}\right)^t. \tag{7}$$

Solving the system (7) has the same complexity as solving (5), but it has to be done only once for each functional I, independent of the number of parameters.

So far, the procedure employed in TRACE is as follows: First, a variation of the geometry is translated into a deformed computational grid, either by generating a new grid starting from the disturbed geometry or by applying a deformation procedure to the given grid for the original geometry. Then, the grid deformation δx in each point is calculated as the difference between new and old coordinates, and the right hand side of (5) is approximated using a (central) finite difference

$$\frac{\partial R}{\partial x}\delta x \approx \frac{R(q, x + h\delta x) - R(q, x - h\delta x)}{2h}, \tag{8}$$

where x denotes the original grid coordinates. This approach has two drawbacks. First, it requires two additional evaluations of the residual, which means that – besides the adjoint solution – the underlying flow solution and details about the solution procedure have to be known for the sensitivity calculation. More important is the fact that for each parameter variation a deformed (three-dimensional) grid has to be generated, stored, and processed to obtain δx. For applications with many parameters, which are of course particularly attractive for employing the adjoint method, this can become a real bottleneck. Therefore it is desirable to reduce the necessary effort for each parameter by adjoining parts of the described pre-process.

The first step, which resolves the issue of additional residual evaluations, is to compute from the adjoint solution the sensitivity of the functional with respect to the mesh coordinates ("mesh sensitivities", see Section 2.1). This corresponds to evaluating $\frac{\partial R}{\partial x}\frac{dx}{d\alpha}$ not as one finite difference, but computing each factor separately. The sensitivity with respect to α is then given by

$$\frac{dI}{d\alpha} = \left(-\psi^t \frac{\partial R}{\partial x}\right)\frac{dx}{d\alpha} = \frac{dI}{dx}\frac{dx}{d\alpha}. \tag{9}$$

For practical computations $\frac{dx}{d\alpha}$ is approximated by the mesh deformation δx, which means that the sensitivity can be obtained by simply computing the scalar product of the mesh sensitivity with the mesh deformation.

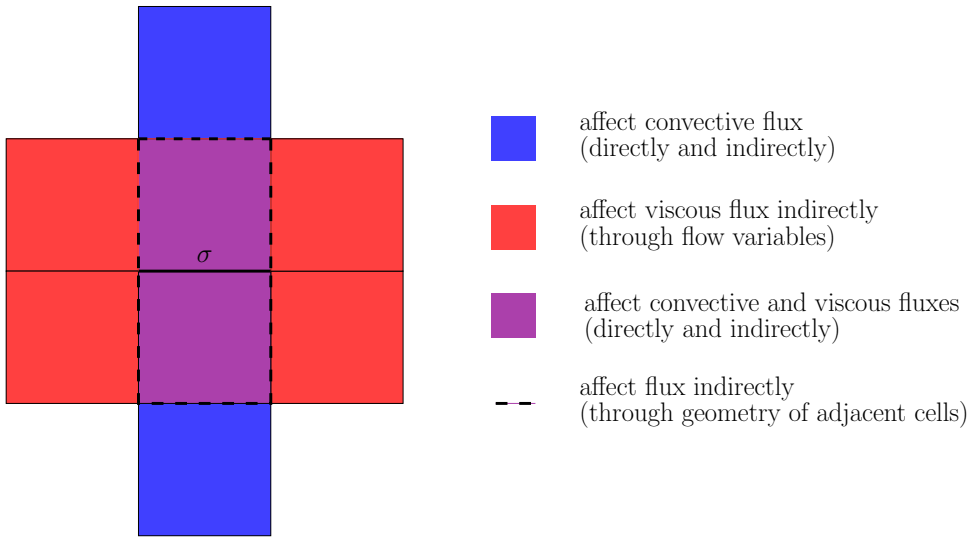

Figure 2: Schematic illustration of the cells and faces which affect with their geometry the flux through the face σ.

To get rid of the need to deal with deformed 3D grids, we have to go one step further and transform the three-dimensional field of mesh sensitivities into a two-dimensional quantity. This is achieved by applying the adjoint mesh deformation procedure described in Section 2.2.

2.1 Computation of mesh sensitivities

In this section we discuss the computation of $\frac{\partial R}{\partial x}$. For a finite volume discretization the residual R_i in cell i is given by

$$R_i = \frac{1}{V_i} \sum_{\sigma \in \partial i} F_\sigma - S_i, \tag{10}$$

where V_i is the volume of cell i, ∂i denotes its boundary (i.e. the set of faces), F_σ is the flux out of cell i through the face σ, and S_i is the source term in i due to rotational effects.

The grid coordinates do not appear in the discretized equations explicitly. In turn, other geometrical quantities which depend on these coordinates (e.g. coordinates of cell centers, cell volumes, face normals) enter the residual expression in many different places, in particular in the flux terms. Therefore it is convenient to split up the computation into two steps. First, the derivative of the residual with respect to all relevant geometrical quantities is evaluated. While the rotational source terms are rather simple and can be differentiated analytically, the derivatives of the fluxes are approximated by finite differences. The geometrical quantities which have to be taken into account can be grouped into cell data and face data. Each flux, and therefore each cell residual, is affected only by the geometry of a limited number of cells and faces (see Fig. 2). This can be exploited to reduce the computational effort. Each contribution is immediately multiplied by the adjoint solution in the corresponding cell to yield a sensitivity which is then stored.

In the second step the derivatives of the intermediate quantities (i.e. cell volumes etc.) with respect to the grid point coordinates have to be evaluated. Since these quantities are given by simple algebraic expressions in the coordinates (see e.g. [20]), the exact derivatives can be written down explicitly. For each point, the contributions for each adjacent cell and face are multiplied by the previously stored sensitivities and then added up. As result, one gets for each vertex the sensitivity of the (total) residual with respect to its three coordinates.

Due to the employed domain decomposition special care has to be taken for the boundary points between two (or more) blocks. Each "copy" of these points contains only the sensitivities for the respective block. For a correct representation of the mesh sensitivities (e.g. for visualization) these values have to be added up. This can be done in a separate post-processing step.

2.2 Adjoint mesh deformation

2.2.1 The mesh deformation algorithm

The elliptic mesh deformation procedure we discuss here is implemented in a pre-processing tool [21]. The deformation vector δx is given as the solution of the Poisson equation

$$\nabla \cdot (E(x)\nabla(\delta x)) = 0, \tag{11}$$

where the modulus E is proportional to the inverse cell volume. More precisely, the deformation of the vertex coordinates δx is given as an interpolation of the cell-centred deformation field,

$$\delta x = J\delta x_{\mathrm{c}}, \tag{12}$$

where the latter solve a cell-centred finite volume discretization of Eqn. (11)

$$R(x_{\mathrm{c}}) = 0. \tag{13}$$

A given surface deformation δy defines an inhomogeneous boundary condition for Eqn. (11), where y denotes the vertex coordinates of the surface mesh. Denoting by $\delta x_{\mathrm{c}}^{\mathrm{int}}$ and $\delta x_{\mathrm{c}}^{\mathrm{ext}}$ the deformation at inner and ghost cells, respectively, the boundary condition is implemented as an extrapolation

$$\delta x_{\mathrm{c}}^{\mathrm{ext}} = T\delta x_{\mathrm{c}}^{\mathrm{int}} + T_{\mathrm{bd}}\delta y, \tag{14}$$

such that the deformation δx, when interpolated onto the surface, coincides with δy. The operators T and T_{bd} depend on the type of boundary condition specified. Apart from Dirichlet boundary conditions, a slip boundary condition has been implemented for boundaries which are not to be deformed but whose surface grid points are allowed to move tangentially along the boundary [21].

From Eqns. (13) and (14) we infer that the cell-centred deformation field is obtained by solving

$$\begin{pmatrix} \dfrac{\partial R}{\partial x_{\mathrm{c}}^{\mathrm{int}}} & \dfrac{\partial R}{\partial x_{\mathrm{c}}^{\mathrm{ext}}} \end{pmatrix} \begin{pmatrix} \mathbf{1} \\ T \end{pmatrix} \delta x_{\mathrm{c}}^{\mathrm{int}} = -\frac{\partial R}{\partial x_{\mathrm{c}}^{\mathrm{ext}}} T_{\mathrm{bd}}\delta y. \tag{15}$$

2.2.2 Adjoint approach

We want to determine the sensitivity of a functional I with respect to a parameter α under the assumption that a variation in α is first transformed into a variation of surface coordinates δy,

$$\delta I = \frac{dI}{d\alpha}\delta\alpha = \frac{dI}{dy}\delta y. \tag{16}$$

Now I depends on y via the 3D mesh deformation δx, i.e.

$$\frac{dI}{dy} = \frac{dI}{dx}\frac{dx}{dy} = \frac{dI}{dx}J\frac{dx_{\mathrm{c}}}{dy}, \tag{17}$$

and δx_c is determined from δy using Eqns. (15) and (14). Hence it follows that

$$\frac{dx_\mathrm{c}}{dy} = \left(\frac{dx_\mathrm{c}^\mathrm{int}}{dy}, \frac{dx_\mathrm{c}^\mathrm{ext}}{dy}\right) = \begin{pmatrix}\mathbf{1}\\ T\end{pmatrix}\frac{dx_\mathrm{c}^\mathrm{int}}{dy} + \begin{pmatrix}\mathbf{0}\\ T_\mathrm{bd}\end{pmatrix}. \tag{18}$$

Denoting by A the system matrix of Eqn. (15), i.e.

$$A = \begin{pmatrix}\frac{\partial R}{\partial x_\mathrm{c}^\mathrm{int}} & \frac{\partial R}{\partial x_\mathrm{c}^\mathrm{ext}}\end{pmatrix}\begin{pmatrix}\mathbf{1}\\ T\end{pmatrix}, \tag{19}$$

it follows that

$$\frac{dx_\mathrm{c}}{dy} = -\begin{pmatrix}\mathbf{1}\\ T\end{pmatrix}A^{-1}\frac{\partial R}{\partial x_\mathrm{c}^\mathrm{ext}}T_\mathrm{bd} + \begin{pmatrix}\mathbf{0}\\ T_\mathrm{bd}\end{pmatrix}. \tag{20}$$

Inserting this equation into Eqn. (17), we obtain

$$\begin{aligned}\frac{dI}{dy} &= \frac{dI}{dx}J\left(-\begin{pmatrix}\mathbf{1}\\ T\end{pmatrix}A^{-1}\frac{\partial R}{\partial x_\mathrm{c}^\mathrm{ext}}T_\mathrm{bd} + \begin{pmatrix}\mathbf{0}\\ T_\mathrm{bd}\end{pmatrix}\right)\\ &= \xi^t\frac{\partial R}{\partial x_\mathrm{c}^\mathrm{ext}}T_\mathrm{bd} + \frac{dI}{dx}J\begin{pmatrix}\mathbf{0}\\ T_\mathrm{bd}\end{pmatrix},\end{aligned} \tag{21}$$

where ξ is given by

$$\xi^t = -\frac{dI}{dx}J\begin{pmatrix}\mathbf{1}\\ T\end{pmatrix}A^{-1}. \tag{22}$$

This definition for ξ can be rewritten as the linear equation

$$A^t\xi = -\begin{pmatrix}\mathbf{1} & T^t\end{pmatrix}J^t\left(\frac{dI}{dx}\right)^t. \tag{23}$$

Once ξ has been computed, the surface sensitivities $\frac{dI}{dy}$ are computed as in Eqn. (21). This is done only once for each cost functional I. The actual sensitivity for a given surface deformation is computed as a surface scalar product

$$\delta I = \frac{dI}{dy}\delta y. \tag{24}$$

It follows that the adjoint mesh deformation allows one to avoid costly 3D mesh deformations for each design parameter.

2.2.3 Implementation of the adjoint deformation

Since the matrix $\begin{pmatrix}\frac{\partial R}{\partial x_\mathrm{c}^\mathrm{int}} & \frac{\partial R}{\partial x_\mathrm{c}^\mathrm{ext}}\end{pmatrix}$ is symmetric, we can use it also for the adjoint deformation. Only the modification due to the boundary conditions (i.e. the operator T) has to be considered separately. Further differences between the adjoint and the original deformation procedure occur in setting up the right hand side of the equation system and in the post-processing of the obtained solution. These steps are in some sense exchanged.

The right hand side is in principle given by the mesh sensitivities $\frac{dI}{dx}$ calculated before, but with some modifications (represented by the operator T and the interpolation J in Eqn. (23)). First, the vertex-based values are interpolated onto cell centers. The obtained values on ghost cells then have to be transferred to inner cells ("adjoint extrapolation"). On block cuts this

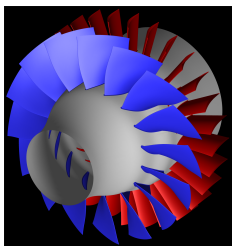

Figure 3: Computational model of the Darmstadt Transonic Compressor.

is achieved by applying an adjoint communication procedure, while on proper boundaries the respective boundary conditions have to be taken into account.

The adjoint solution ξ defined on cells has to be transformed into surface values. This corresponds to the application of $\frac{\partial R}{\partial x_c^{\text{ext}}} T_{\text{bd}}$ (cf. Eqn. (21)) from the right and is adjoint to the computation of the right hand side of Eqn. (13), where (inner) cell values are determined from the surface deformation. The second summand in Eqn. (21) corresponds to taking the value of the right hand side of the adjoint system in the ghost cell, i.e. the right hand side of Eqn. (23) without the application of the matrix $\begin{pmatrix} \mathbf{1} & T^t \end{pmatrix}$, and applying T_{bd} to it.

2.3 Results

As a test case for the validation of our implementation we use the Darmstadt Transonic Compressor with the baseline rotor geometry (Rotor 1) [22], see Fig. 3. We consider an operating point at a rotor velocity of 20,000 rounds per minute and a mass flow of 16.2 kg/s. The pre-shock Mach number is approximately 1.5.

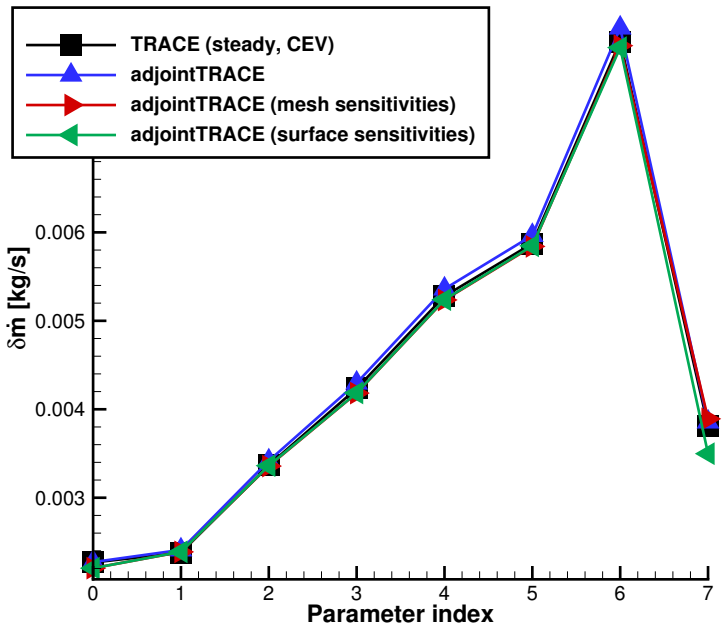

Figure 4: Sensitivities of mass flow obtained with the nonlinear solver and different processes using the adjoint solver.

We compare the sensitivities obtained using the adjoint solver with the three different methods described above to (central) finite differences of nonlinear steady solutions. For the latter, the constant eddy viscosity (CEV) assumption has been employed, i.e. the eddy viscosity for the initial geometry is used also for the calculations on the deformed grids and kept fixed during these calculations. This is necessary for getting comparable results, since the adjoint solver does not take into account the turbulence model. We use mass flow at the exit as objective functional. The geometric parameters are small variations of the stagger angle at eight different radial heights, as described in [17]. The results are shown in Fig. 4. We observe a good agreement of all sensitivities. In particular, the sensitivities using mesh sensitivities and those using surface sensitivities agree almost exactly, except for the last parameter. This deviation is due to a subtle difference between the forward and the adjoint deformation procedure. After the solution of the deformation problem has been obtained and interpolated onto the grid vertices, an additional correction procedure is applied at block cuts, which was not taken into account in the implementation of the adjoint deformation.

As an example for the resulting surface sensitivities we show the suction side of the rotor blade in Fig. 5. The largest sensitivities can be seen at the leading and trailing edge. As the pressure distribution shows, another region of rather high sensitivities occurs near the shock in the upper part of the blade.

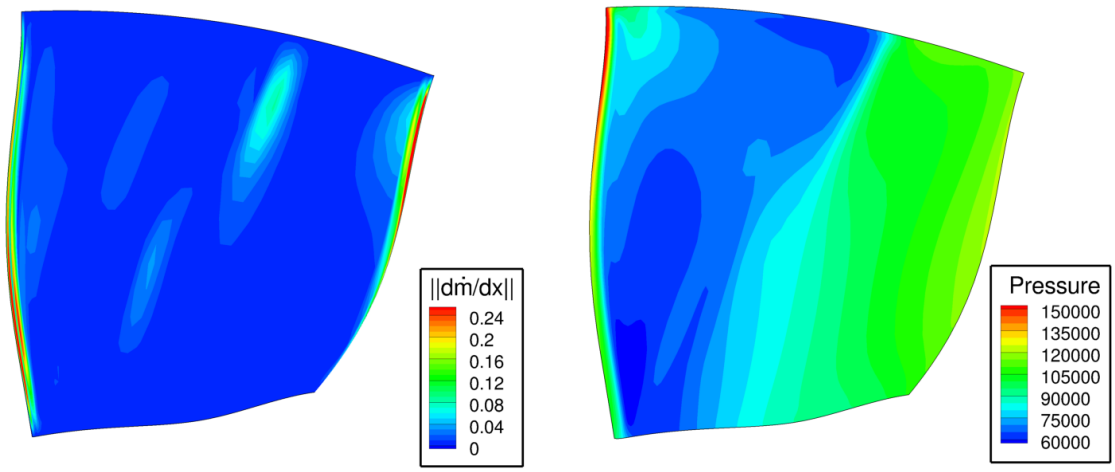

Figure 5: Left: Norm of the sensitivity vector of exit mass flow with respect to surface grid coordinates for the suction side of the rotor blade of the Darmstadt Transonic Compressor. Right: Pressure distribution on the blade.

In the main adjoint process, which is aimed at computing sensitivities with respect to design parameters, the 3D mesh sensitivities are only an intermediate result. But they also provide interesting information by themselves. For a good mesh one would expect that the sensitivities in the inner of the domain, i.e. away from solid boundaries, are zero. If in some area there are large sensitivities, this is an indication that the mesh should be refined there to properly resolve the flow solution. To illustrate this, we performed, for the same test case as before, calculations on two different meshes. In Fig. 6 the x component of the mesh sensitivities is shown as an example. While on the finer mesh it is mostly close to zero – except for a few spots near the leading edge and in the region of the passage shock – we see very large sensitivities (about a factor of ten larger) for the coarse mesh, in particular in the passage.

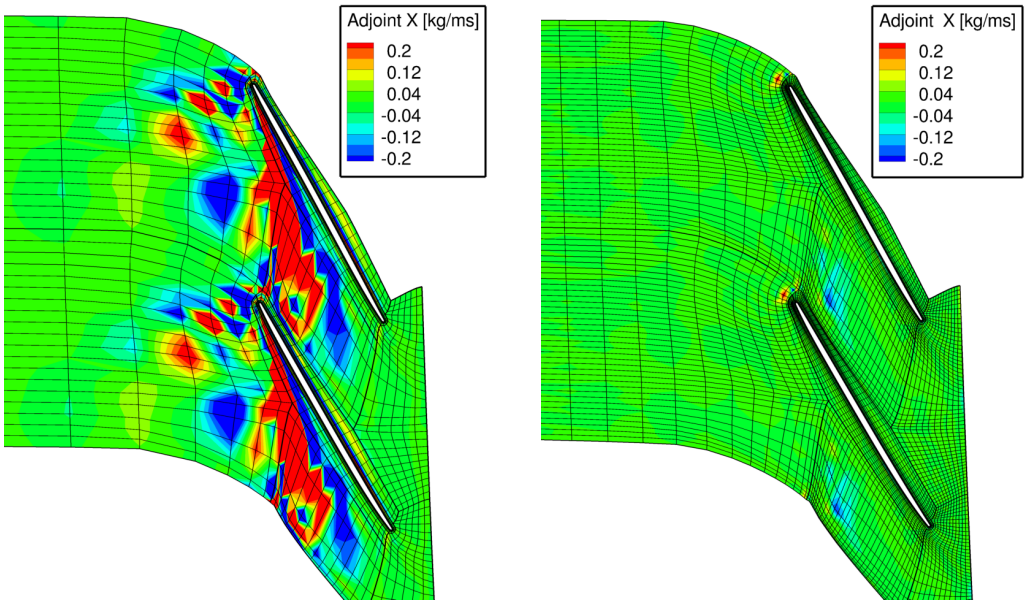

Figure 6: Mesh sensitivities (x component) of exit mass flow in the rotor row of the Darmstadt Transonic Compressor for two different meshes.

3 SENSITIVITIES WITH RESPECT TO BOUNDARY VALUES

In this section we first discuss briefly the implementation of nonreflecting entry and exit boundary conditions for the linear and adjoint solver, as well as an extension which allows to use inhomogeneous boundary conditions in linear calculations (see Section 3.1). This is used to compute sensitivities with respect to boundary values, i.e. those quantities which are prescribed at entries and exits, respectively. We denote the set of boundary values by q_{bv}. At outlets, this consists only of the (static) pressure, while for entries it contains four variables, e.g. entropy, stagnation enthalpy, and two flow angles. Typically, the user provides inlet conditions for the stagnation pressure and temperature. Entropy and stagnation enthalpy are then calculated from these boundary values.

The evaluation of sensitivities with respect to the boundary values q_{bv} using the linear and the adjoint approach is described in Section 3.2. In the following part (Section 3.3) we discuss the extension of the exit boundary conditions for the radial equilibrium.

3.1 Entry and exit boundary conditions in the linear and adjoint solver

In general, boundary conditions in TRACE are implemented using the concept of ghost cells. This means that we divide the set of flow field variables into the inner and the ghost cell part: $q = (q^{\mathrm{int}}, q^{\mathrm{ext}})$. The values in the inner cells are determined as solution of the flow equations, while the ghost cell values are given as a function of the inner values and prescribed boundary values:

$$q^{\mathrm{ext}} = \mathcal{T}(q^{\mathrm{int}}, q^{*}_{\mathrm{bv}}). \tag{25}$$

For the linear solver a linearized update operator $T = \dfrac{\partial \mathcal{T}}{\partial q^{\mathrm{int}}}$ is implemented. The linearization of the residual with respect to the inner cell values then reads

$$\delta q^{\mathrm{int}} \mapsto \frac{\partial R}{\partial q} \begin{pmatrix} \mathbf{1} \\ T \end{pmatrix} \delta q^{\mathrm{int}}. \tag{26}$$

In the adjoint solver the left-hand side is the adjoint of (26)

$$\begin{pmatrix} \mathbf{1} & T^t \end{pmatrix} \left(\frac{\partial R}{\partial q} \right)^t . \tag{27}$$

Hence, after each matrix-vector multiplication an adjoint boundary update is performed that adds $T^t \psi^{\text{ext}}$ to ψ^{int}. For a more detailed description of the boundary treatment in the non-linear, linear, and adjoint solver, the reader is referred to [23].

Consider the case of an entry or exit and assume that q^*_{bv} is kept fix. The linearization of the update operator in (25) can be expressed in terms of characteristic variables as follows. The outgoing characteristics are copied to the ghost cells from the interior cells, i.e. $\delta c^{\text{ext}}_{\text{out}} = \delta c^{\text{int}}_{\text{out}}$. Then the incoming characteristic variables in the ghost cells are modified in such a way that the variations of the specified boundary values vanish. Define the boundary values as the corresponding function of the averaged conservative variables of q^{int} and q^{ext}, i.e.

$$q_{\text{bv}} = q_{\text{bv}}(\tfrac{1}{2}(q^{\text{int}} + q^{\text{ext}})). \tag{28}$$

Writing the linearization of (28) in terms of characteristic variables and solving for the variation of the incoming characteristics, one obtains

$$\delta c^{\text{ext}}_{\text{inc}} = -\delta c^{\text{int}}_{\text{inc}} - 2 \left(\frac{\partial q_{\text{bv}}}{\partial c_{\text{inc}}} \right)^{-1} \frac{\partial q_{\text{bv}}}{\partial c_{\text{out}}} \delta c^{\text{int}}_{\text{out}}. \tag{29}$$

For entries, Eqn. (29) specializes to

$$\delta c^{\text{ext}}_k = -\delta c^{\text{int}}_k - 2 \left(\left(\frac{\partial q_{\text{bv}}}{\partial c_{1\ldots4}} \right)^{-1} \frac{\partial q_{\text{bv}}}{\partial c_5} \right)_k \delta c^{\text{int}}_5 \quad \text{for } k = 1, \ldots, 4 \tag{30}$$

and for exits we get

$$\delta c^{\text{ext}}_5 = -\delta c^{\text{int}}_5 - 2 \left(\frac{\partial p}{\partial c_5} \right)^{-1} \sum_{k=1}^{4} \frac{\partial p}{\partial c_k} \delta c^{\text{int}}_k . \tag{31}$$

Equation (29) (or Eqn. (30) or (31), respectively) is applied to states averaged over one band, where a band consists of all boundary faces at (approximately) the same radius. The result is an average state to be prescribed at the exterior cells. Several types of average techniques are used, e.g. mass or flux averages [24], which means that (28) is slightly generalized. For the treatment of different types of averages and the integration into the non-reflecting boundary condition, see [23].

In the adjoint solver, for each routine of the linear boundary update an adjoint routine has been implemented, and the order of the steps is reversed. In particular, adjoint versions of equations (30) and (31) are required. For this we have to determine the derivative of the characteristic variables in the ghost cells with respect to those in the inner:

$$\chi^{\text{int}} = \frac{\partial(\delta c^{\text{ext}})}{\partial(\delta c^{\text{int}})} \chi^{\text{ext}}, \tag{32}$$

where we denote the adjoint characteristic variables by χ. For entries we get

$$\chi^{\text{int}}_k = -\chi^{\text{ext}}_k \text{ for } k = 1, \ldots, 4 \tag{33}$$

and

$$\chi_5^{\text{int}} = \chi_5^{\text{ext}} - 2\sum_{k=1}^{4}\left(\left(\frac{\partial q_{\text{bv}}}{\partial c_{1\ldots4}}\right)^{-1}\frac{\partial q_{\text{bv}}}{\partial c_5}\right)_k \chi_k^{\text{ext}} \tag{34}$$

and for exits (32) yields

$$\chi_k^{\text{int}} = \chi_k^{\text{ext}} - 2\left(\frac{\partial p}{\partial c_5}\right)^{-1}\frac{\partial p}{\partial c_k}\chi_5^{\text{ext}} \text{ for } k = 1,\ldots,4 \tag{35}$$

and

$$\chi_5^{\text{int}} = -\chi_5^{\text{ext}}. \tag{36}$$

For the linear solver we also implemented inhomogeneous boundary conditions. This means that we replace the condition $\delta q_{\text{bv}} = 0$ by $\delta q_{\text{bv}} = \delta q_{\text{bv}}^*$, where δq_{bv}^* is some prescribed value. This leads to modifications of equations (30) and (31), namely

$$\delta c_k^{\text{ext}} = -\delta c_k^{\text{int}} - 2\left(\left(\frac{\partial q_{\text{bv}}}{\partial c_{1\ldots4}}\right)^{-1}\frac{\partial q_{\text{bv}}}{\partial c_5}\right)_k \delta c_5^{\text{int}} + 2\left(\left(\frac{\partial q_{\text{bv}}}{\partial c_{1\ldots4}}\right)^{-1}\delta q_{\text{bv}}^*\right)_k \tag{37}$$

for entries and

$$\delta c_5^{\text{ext}} = -\delta c_5^{\text{int}} - 2\left(\frac{\partial p}{\partial c_5}\right)^{-1}\sum_{k=1}^{4}\frac{\partial p}{\partial c_k}\delta c_k^{\text{int}} + 2\left(\frac{\partial p}{\partial c_5}\right)^{-1}\delta p^* \tag{38}$$

for exits.

3.2 Implementation of the sensitivity evaluation

We assume now that the prescribed boundary values depend on a parameter α and we want to compute the sensitivity of a functional I with respect to α. First we consider the linear approach. For this we use the inhomogeneous boundary conditions given by Eqns. (37) and (38). These correspond to the linearization of the boundary operator $\mathcal{T}$ (cf. Eqn. (25)) with respect to δq_{bv}^*, which we denote by T_{bv}. Using this we get for the derivative of q with respect to α

$$\frac{dq}{d\alpha} = \left(\frac{dq^{\text{int}}}{d\alpha}, \frac{dq^{\text{ext}}}{d\alpha}\right) = \left(\frac{dq^{\text{int}}}{d\alpha}, T\frac{dq^{\text{int}}}{d\alpha} + T_{\text{bv}}\frac{dq_{\text{bv}}^*}{d\alpha}\right) = \begin{pmatrix}\mathbf{1}\\ T\end{pmatrix}\frac{dq^{\text{int}}}{d\alpha} + \begin{pmatrix}\mathbf{0}\\ T_{\text{bv}}\frac{dq_{\text{bv}}^*}{d\alpha}\end{pmatrix}. \tag{39}$$

Inserting this into (3) yields

$$\frac{dI}{d\alpha} = \frac{\partial I}{\partial q}\begin{pmatrix}\mathbf{1}\\ T\end{pmatrix}\frac{dq^{\text{int}}}{d\alpha} + \frac{\partial I}{\partial q^{\text{ext}}}T_{\text{bv}}\frac{dq_{\text{bv}}^*}{d\alpha}. \tag{40}$$

Similarly, we can rewrite Eqn. (5), where we have in this case $\frac{dx}{d\alpha} = 0$, and get

$$\frac{\partial R}{\partial q}\begin{pmatrix}\mathbf{1}\\ T\end{pmatrix}\frac{dq^{\text{int}}}{d\alpha} = -\frac{\partial R}{\partial q^{\text{ext}}}T_{\text{bv}}\frac{dq_{\text{bv}}^*}{d\alpha}. \tag{41}$$

In our implementation, the right hand side of (41) is computed by applying the inhomogeneous boundary condition to a zero vector (which changes only the ghost cell values and leaves all values on interior cells equal to zero) and multiplying the result by the system matrix $\frac{\partial R}{\partial q}$. The sensitivity evaluation is slightly different from what is suggested by Eqn. (40), since the two summands are not computed separately. Instead, we rewrite the equation as

$$\frac{dI}{d\alpha} = \frac{\partial I}{\partial q}\left(\begin{pmatrix}\mathbf{1}\\ T\end{pmatrix}\frac{dq^{\text{int}}}{d\alpha} + \begin{pmatrix}\mathbf{0}\\ T_{\text{bv}}\dfrac{dq^*_{\text{bv}}}{d\alpha}\end{pmatrix}\right). \tag{42}$$

The expression in parenthesis is obtained by applying the linear boundary operator again to the solution of (41) and the sensitivity is computed as scalar product of the resulting vector with the linearized functional $\frac{\partial I}{\partial q}$.

For the adjoint solver, the sensitivity evaluation is implemented as a post-processing step during the determination of the adjoint solution. It is carried out each time the (approximate) solution vector is available, i.e. at each restart of the GMRES algorithm. The sensitivity is in this case given by

$$\frac{dI}{d\alpha} = -\psi^t \frac{\partial R}{\partial q^{\text{ext}}} T_{\text{bv}} \frac{dq^*_{\text{bv}}}{d\alpha} + \frac{\partial I}{\partial q^{\text{ext}}} T_{\text{bv}} \frac{dq^*_{\text{bv}}}{d\alpha} = \left(-\psi^t \frac{\partial R}{\partial q^{\text{ext}}} + \frac{\partial I}{\partial q^{\text{ext}}}\right) T_{\text{bv}} \frac{dq^*_{\text{bv}}}{d\alpha}, \tag{43}$$

where ψ is the solution of

$$\left(\frac{\partial R}{\partial q}\begin{pmatrix}\mathbf{1}\\ T\end{pmatrix}\right)^t \psi = \left(\frac{\partial I}{\partial q}\begin{pmatrix}\mathbf{1}\\ T\end{pmatrix}\right)^t = \begin{pmatrix}\mathbf{1} & T^t\end{pmatrix}\left(\frac{\partial I}{\partial q}\right)^t. \tag{44}$$

No modification of the actual adjoint code, e.g. boundary conditions, is necessary since existing routines can be used to compute the terms appearing in Eqn. (43). The second factor, $T_{\text{bv}}\frac{dq_{\text{bv}}}{d\alpha}$, is the same that appears on the right hand side of the linear system (41). The product in the first term is rewritten as $\left(\frac{\partial R}{\partial q^{\text{ext}}}\right)^t \psi$, which is nothing else than the system matrix of the adjoint problem applied to the adjoint solution. Here, $\frac{\partial R}{\partial q^{\text{ext}}}$ can be replaced by the full matrix $\frac{\partial R}{\partial q}$, since the subsequent scalar product is evaluated on the ghost cells only. For the same reason, $\frac{\partial I}{\partial q}$, which is known from the computation of the right hand side of the adjoint system (44), can be used instead of $\frac{\partial I}{\partial q^{\text{ext}}}$.

3.3 Incorporation of radial equilibrium

So far we have dealt only with boundary conditions where the boundary values are prescribed band-wise, with previously fixed values. But in turbomachinery applications it is common that the pressure distribution is not known in advance, but determined by the so called radial equilibrium condition. This applies to rotationally symmetric flows and is given by the differential equation

$$\frac{\partial p}{\partial r} = \rho r (U^\theta + \omega)^2 \tag{45}$$

in a rotating frame of reference, where ω is the rotation speed. It is used in the exit boundary condition of the nonlinear solver in the form that an average pressure at a certain position ("reference band"), e.g. at midspan, is prescribed and then integration towards hub and tip is performed using the following discretization of Eqn. (45)

$$\frac{p_n - p_{n-1}}{r_n - r_{n-1}} = \frac{1}{2}(f_n + f_{n-1}), \tag{46}$$

where $f_n = \rho_n r_n (U_n^\theta + \omega)^2$.

For consistency with the underlying nonlinear solver it is desirable to provide a counterpart of this boundary condition also in the linear and adjoint solvers. In the linear solver, this is achieved by employing a similar idea as for the inhomogeneous boundary conditions, namely to modify the way in which the characteristic values in the ghost cells are determined from those in the inner cells.

For the reference band, the boundary condition (either the homogeneous or the inhomogeneous form) is applied as described in Section 3.1. For the other bands, the new ghost cell values are determined iteratively from the condition that the solution at the boundary, $\delta q^{\mathrm{bd}} = \frac{1}{2}(\delta q^{\mathrm{int}} + \delta q^{\mathrm{ext}})$, satisfies the linearization of (46), i.e.

$$\frac{\delta p_n - \delta p_{n-1}}{r_n - r_{n-1}} = \frac{1}{2}(\delta f_n + \delta f_{n-1}), \tag{47}$$

where $\delta f_n = r_n (U_n^\theta + \omega)^2 \delta\rho_n + 2\rho_n r_n (U_n^\theta + \omega)\delta U_n^\theta$. This can again be expressed in terms of characteristic variables, yielding $\delta c_5^{\mathrm{ext}} = -\delta c_5^{\mathrm{int}} + 2\delta c_5^{\mathrm{bd}}$ with

$$\begin{aligned}\delta c_{5,n}^{\mathrm{bd}} = \Bigg(&- \frac{\partial p}{\partial c_4}\delta c_{4,n} + \tfrac{1}{2}\Delta r_n r_n (U_n^\theta + \omega)^2 \Big(\frac{\partial \rho}{\partial c_1}\delta c_{1,n} + \frac{\partial \rho}{\partial c_4}\delta c_{4,n} \Big) + \Delta r_n r_n \rho_n (U_n^\theta + \omega)\delta U_n^\theta \\ &+ \delta p_{n-1} + \tfrac{1}{2}\Delta r_n \delta f_{n-1} \Bigg) \Big/ \Bigg(\frac{\partial p}{\partial c_5} - \tfrac{1}{2}\Delta r_n r_n (U_n^\theta + \omega)^2 \frac{\partial \rho}{\partial c_5} \Bigg),\end{aligned} \tag{48}$$

where we use the abbreviation $\Delta r_n := r_n - r_{n-1}$ and leave out the superscript "bd" for all quantities on the right hand side.

The implementation of the adjoint radial equilibrium condition based on this linearization is analogous to that of the "usual" adjoint boundary condition (at exits) described in Section 3.1.

3.4 Results

As a first test case for the validation of our implementation we use a 2D section of a subsonic turbine stage. First, we test the linearized version by comparing the solution obtained with the linear solver using inhomogeneous boundary conditions with the difference of two solutions calculated by the nonlinear solver with different boundary conditions. An example for the variation of the exit pressure is shown in Fig. 7, where a very good agreement can be observed.

The corresponding sensitivities – normalized to a change of 1 Pa – for some objective functions are collected in Table 1 together with those obtained by the adjoint solver. For all functionals except mass flow we see only very small (relative) differences between the three sensitivities. In particular, the linear and adjoint results agree very well, which shows the correctness of the adjoint implementation.

functional	mass flow	pressure ratio	total pressure ratio	total temperature ratio
nonlinear	-6.0466e-03	0.5996	0.3539	0.1082
linear	-6.0752e-03	0.5997	0.3546	0.1090
adjoint	-6.0994e-03	0.5995	0.3544	0.1093

Table 1: Sensitivities of several objective functions with respect to static pressure at exit for a subsonic turbine stage obtained with the nonlinear, linear, and adjoint solver, respectively. The results are normalized with the reference outlet mass flow and pressure.

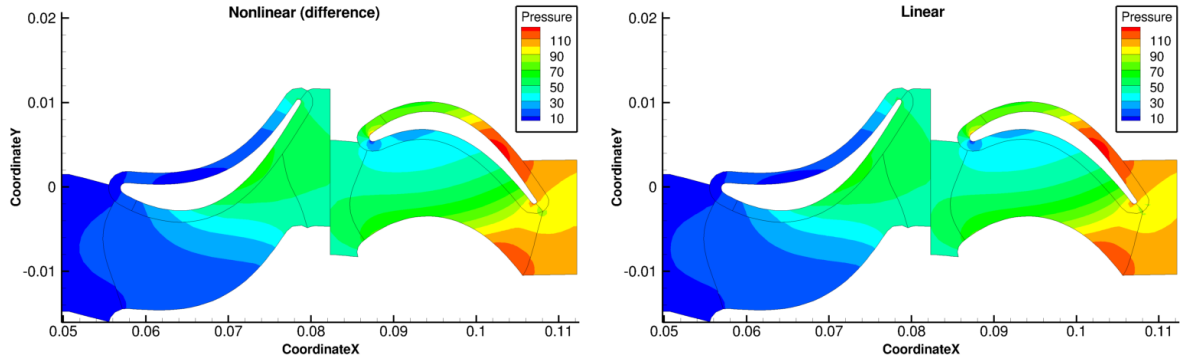

Figure 7: Comparison of flow solutions for a subsonic turbine stage. Left: Difference of two nonlinear solutions where the static pressure at the outlet differs by 100 Pa. Right: Linear solution where a pressure difference of 100 Pa has been prescribed at the outlet.

We did a similar comparison for the Darmstadt Transonic Compressor described in Section 2.3. Here we did the calculations only for the rotor row. The results are collected in Table 2 and show again an excellent agreement between linear and adjoint sensitivities. The deviations with respect to the nonlinear results are larger, but still acceptable – at most about five percent. In most cases this can be improved by replacing the one-sided finite differences with central differences.

functional	mass flow	pressure ratio	total pressure ratio	total temperature ratio
nonlinear	-1.3388	0.7525	1.1093	0.3393
linear	-1.2689	0.7857	1.1108	0.3235
adjoint	-1.2688	0.7857	1.1108	0.3235

Table 2: Sensitivities of several objective functions with respect to static pressure at exit for the Darmstadt Compressor obtained with the nonlinear, linear, and adjoint solver, respectively. A constant pressure difference was prescribed at the outlet. The results are normalized with the reference outlet mass flow and pressure.

Since the sensitivities with respect to boundary values are – if the adjoint solver is used – computed at each restart (see Section 3.2) we can also analyze their convergence behaviour. In Fig. 8 this is shown for two functionals for the turbine test case from above. Also shown is the residual which indicates the convergence of the adjoint solution. In both cases we observe that the sensitivity does not change significantly after the residual has dropped below 10^{-4}.

To test our implementation of the linearized and adjoint radial equilibrium condition we also use the rotor of the Darmstadt Compressor as test case. We apply the same deformations as in Section 2.3, which only affect the rotor. First we did calculations without radial equilibrium, i.e. with constant pressure prescribed at the outlet. The corresponding results for two functionals are shown in Fig. 9. While the sensitivities calculated with the linear and adjoint solver agree almost exactly, we observe a small deviation with respect to the nonlinear solver. For mass flow, these differences are somewhat larger than those seen in Fig. 4, where the stator row was included in the calculation. One reason for this could be that prescribing a constant pressure at the outlet of the rotor row is an unphysical boundary condition.

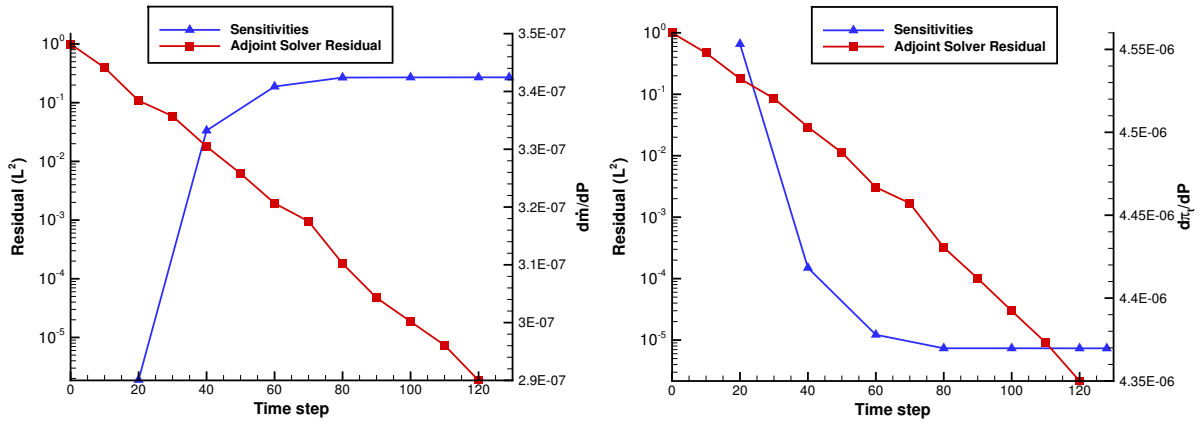

Figure 8: Convergence of sensitivities of mass flow (left) and total pressure ratio (right) with respect to exit pressure and corresponding residuals for a subsonic turbine stage.

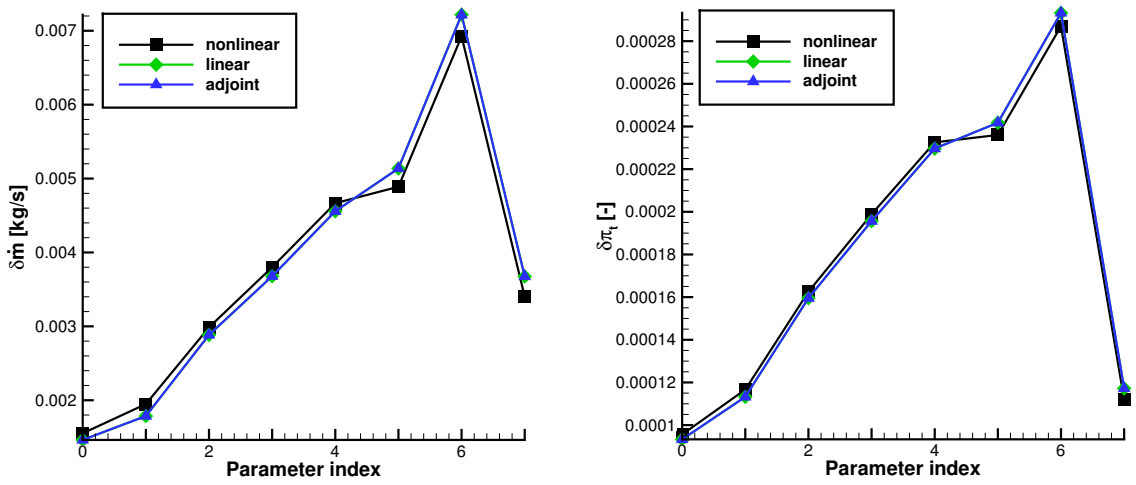

Figure 9: Sensitivities of mass flow (left) and total pressure ratio (right) for the rotor row of the Darmstadt Compressor with constant pressure prescribed at the outlet, calculated with the nonlinear, linear, and adjoint solver.

functional	mass flow	pressure ratio	total pressure ratio	total temperature ratio
nonlinear	-1.3685	0.8142	1.2353	0.3705
linear	-1.5244	0.7630	1.2172	0.3836
adjoint	-1.5244	0.7629	1.2171	0.3836

Table 3: Sensitivities of several objective functions with respect to static pressure at midspan of the outlet (with radial equilibrium condition applied) for the Darmstadt Compressor obtained with the nonlinear, linear, and adjoint solver, respectively. The results are normalized with the reference outlet mass flow and pressure.

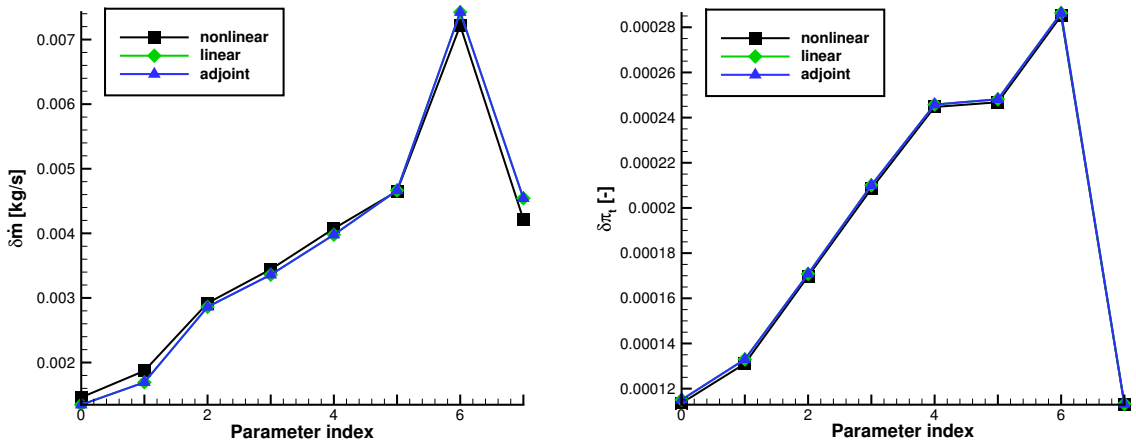

Figure 10: Sensitivities of mass flow (left) and total pressure ratio (right) for the rotor row of the Darmstadt Compressor with radial equilibrium applied at the outlet, calculated with the nonlinear, linear, and adjoint solver.

When we take into account the radial equilibrium (Fig. 10), the picture is similar, but the difference between adjoint/linear and nonlinear sensitivities is in many cases smaller. The good agreement of all values indicates that the linearized and adjoint versions of the radial equilibrium condition are implemented correctly.

The radial equilibrium condition can also be used in the computation of sensitivities with respect to boundary values. In the case of outlet pressure, the obtained sensitivity is that with respect to the pressure at the reference band. We repeated the computations presented in Table 2 using the radial equilibrium condition (cf. Table 3). In this case the prescribed pressure difference applies only for the reference band. The results are very similar to those without radial equilibrium, except that the differences between nonlinear and linear/adjoint sensitivities are in some cases larger, in particular for the mass flow functional.

As a further test we look at the radial distribution of the pressure at the exit. More precisely, we compare the difference between the distributions from the two nonlinear calculations (the same that were used for Table 3) to the distribution obtained from the linear solution, which also corresponds to a pressure difference (cf. Fig. 11). We observe a good agreement of the overall shape of the two curves. The fluctuations seen in the linear solution are similar to those in the distribution from a calculation without radial equilibrium and thus seem not to be due to the implementation of the radial equilibrium condition.

4 CONCLUSIONS

We have discussed two extensions of a solver for the adjoint RANS equations. First we have showed how sensitivities with respect to geometric parameters can be computed more efficiently – for large numbers of parameters – by first computing from the adjoint solution sensitivities with respect to grid coordinates on the surfaces which are affected by the parameters. This corresponds to adjoining the pre-process of the flow solver. As an intermediate result we obtain sensitivities with respect to the coordinates of the three-dimensional computational grid. These can also be used to judge the quality of the grid – at least for the computation of the given objective function – which we have not examined systematically so far.

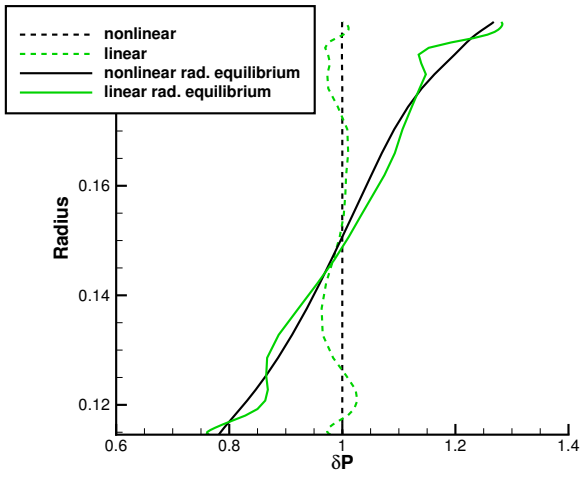

Figure 11: Radial distribution of the pressure difference at the outlet of the rotor row of the Darmstadt Compressor calculated with the linear and nonlinear solver. The dotted lines represent calculations where a constant pressure (difference) was prescribed.

In the second part we have discussed the computation of sensitivities with respect to boundary values which are prescribed at inlets and outlets of turbomachinery configurations. In particular we have focused on the pressure at outlets, since this is relevant e.g. for optimizations where outflow conditions like mass flow are used as a constraint. Moreover we have presented an extension of the boundary conditions for the linear and adjoint solver, namely the radial equilibrium condition. These developments have been validated by representative calculations of turbomachinery components. In particular we have observed an excellent agreement between sensitivities obtained with the linear and the adjoint solver, which shows that the implemented boundary conditions are indeed adjoints of each other.

ACKNOWLEDGEMENTS

The authors would like to thank MTU Aero Engines for their support and co-sponsorship of the first author.

REFERENCES

[1] B. Mohammadi and O. Pironneau, *Applied Shape Optimization for Fluids*. Oxford University Press; Second Edition edition, 2009.

[2] J. E. Peter and R. P. Dwight, "Numerical sensitivity analysis for aerodynamic optimization: A survey of approaches," *Computers & Fluids*, vol. 39, no. 3, pp. 373 – 391, 2010.

[3] K. Giannakoglou, D. Papadimitriou, and I. Kampolis, "Aerodynamic shape design using evolutionary algorithms and new gradient-assisted metamodels," *Computer Methods in Applied Mechanics and Engineering*, vol. 195, no. 44-47, pp. 6312 – 6329, 2006.

[4] M. Bompard, J. Peter, and J.-A. Désidéri, "Surrogate models based on function and derivative values for aerodynamic global optimization," V European Conference on Computational Fluid Dynamics ECCOMAS CFD 2010, Lisbon, Portugal.

[5] J. Backhaus, M. Aulich, C. Frey, T. Lengyel, and C. Voß, "Gradient enhanced surrogate models based on adjoint CFD methods for the design of a counter rotating turbofan." To appear in Proceedings of the ASME-GT2012.

[6] J. Lions, *Optimal control of systems governed by partial differential equations*. Die Grundlehren der mathematischen Wissenschaften. Band 170. Springer-Verlag, Berlin-Heidelberg-New York., 1971.

[7] A. Jameson, "Aerodynamic design via control theory," *Journal of Scientific Computing*, vol. 3, no. 3, pp. 233–260, 1988.

[8] D. Ghate and M. Giles, *Inexpensive Monte Carlo uncertainty analysis*, pp. 203–210. Tata McGraw-Hill, New Delhi, 2006.

[9] M. B. Giles and E. Süli, "Adjoint methods for PDEs: a posteriori error analysis and post-processing by duality," *Acta Numerica*, vol. 11, pp. 145–236, 2002.

[10] D. A. Venditti and D. L. Darmofal, "Grid adaptation for functional outputs: Application to two-dimensional inviscid flows," *Journal of Computational Physics*, vol. 176, no. 1, pp. 40 – 69, 2002.

[11] D. A. Venditti and D. L. Darmofal, "Anisotropic grid adaptation for functional outputs: application to two-dimensional viscous flows," *Journal of Computational Physics*, vol. 187, no. 1, pp. 22 – 46, 2003.

[12] N. R. Gauger, A. Walther, C. Moldenhauer, and M. Widhalm, "Automatic differentiation of an entire design chain for aerodynamic shape optimization," in *New Results in Numerical and Experimental Fluid Mechanics VI* (N. Kroll and J. K. Fassbender, eds.), vol. 96 of *Notes on Numerical Fluid Mechanics and Multidisciplinary Design*, ch. 13, pp. 181–193, Springer Berlin / Heidelberg, 2008.

[13] D. Nürnberger, F. Eulitz, S. Schmitt, and A. Zachcial, "Recent progress in the numerical simulation of unsteady viscous multistage turbomachinery flow," in *ISABE 2001-1081*, 2001.

[14] K. Becker, K. Heitkamp, and E. Kügeler, "Recent progress in a hybrid-grid CFD solver for turbomachinery flows," in *Proceedings Fifth European Conference on Computational Fluid Dynamics ECCOMAS CFD 2010. V European Conference on Computational Fluid Dynamics ECCOMAS CFD 2010*, 2010.

[15] C. Frey, D. Nürnberger, and H. Kersken, "The discrete adjoint of a turbomachinery RANS solver," in *Proceedings of ASME-GT2009*, 2009.

[16] H.-P. Kersken, C. Frey, C. Voigt, and G. Ashcroft, "Time-linearized and time-accurate 3d RANS methods for aeroelastic analysis in turbomachinery," in *ASME TurboExpo 2010*, 2010.

[17] C. Frey, G. Ashcroft, J. Backhaus, E. Kügeler, and J. Wellner, "Adjoint-based flow sensitivity analyis using arbitrary control surfaces," in *Proceedings of ASME-GT2011*.

[18] Z. Yang and D. J. Mavripilis, "Unstructured dynamic meshes with higher-order time integration schemes for the unsteady navier-stokes equations," in *43th AIAA Aerospace Sciences Meeting and Exhibit, AIAA Paper 2005-1222*, (Reno, USA), 2005.

[19] R. Dwight, "Robust mesh deformation using the linear elasticity equations," in *Computational Fluid Dynamics 2006* (H. Deconinck and E. Dick, eds.), pp. 401–406, Springer, 2006.

[20] J. Blazek, *Computational fluid dynamics: principles and applications*. Elsevier Science, 2001.

[21] C. Voigt, C. Frey, and H.-P. Kersken, "Development of a generic surface mapping algorithm for fluid-structure-interaction simulations in turbomachinery," in *V European Conference on Computational Fluid Dynamics ECCOMAS CFD 2010* (J. C. F. Pereira, A. Sequeira, and J. M. C. Pereira, eds.), 2010.

[22] Schulze G., Hennecke D.K., Sieber J., Wöhrl B., "Der neue Verdichterprüfstand an der TH Darmstadt," *VDI Berichte Nr. 1109, Germany.*, 1994.

[23] C. Frey, A. Engels-Putzka, and E. Kügeler, "Adjoint boundary conditions for turbomachinery flows." To appear in Proceedings of the 6th European Congress on Computational Methods in Applied Science and Engineering ECCOMAS 2012, Vienna, Austria.

[24] E. M. Greitzer, C. S. Tan, and M. B. Graf, *Internal Flow*. Cambridge Univ. Press, New York, 2004.